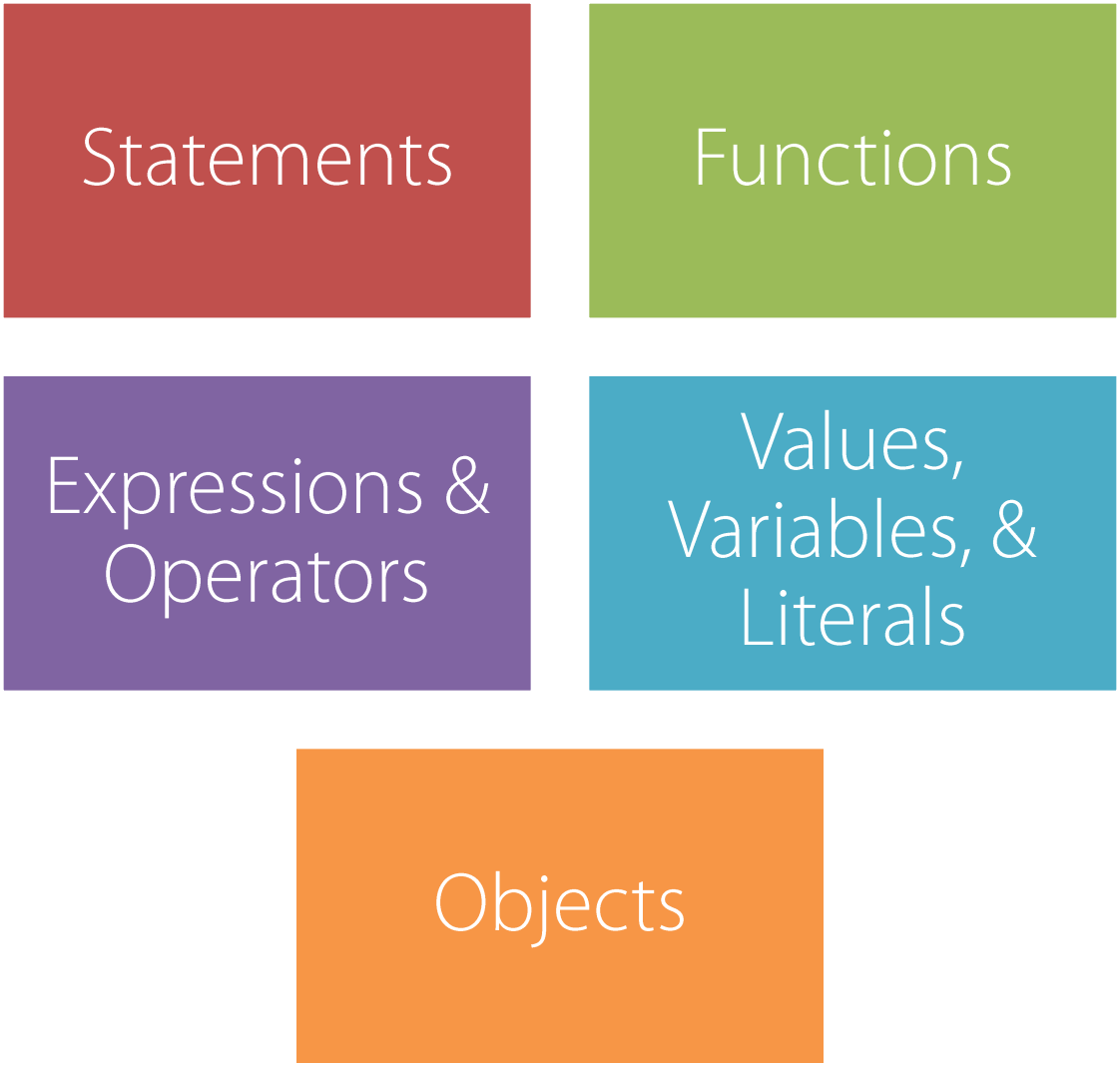
Fixing Common JavaScript Bugs

Elijah Manor
http://elijahmanor.com
@elijahmanor

pluralsight
hardcore developer training


HELLO
ELIJAH
SPAM
Hormel


Statements
Functions
Expressions & Operators
Values, Variables, & Literals
Objects

Fixing Common JavaScript Bugs

Statements

pluralsight
hardcore developer training

Missing Mark Bug

Examine

```
function getNames() {
  var length = 0, names = ""

  ['John', 'Susan', 'Joe'].forEach(function (name, i) {
    length = i + 1
    names += name + ' '
  })

  return
  {
    length: length,
    names: names
  }
}
```

Missing Mark Bug

Expose

```
function getNames() {
  var length = 0, names = ""

  ['John', 'Susan', 'Joe'].forEach(function (name, i) {
    length = i + 1
    names += name + ' '
  })

  return
  {
    length: length,
    names: names
  }
}
```

Uncaught SyntaxError:
Unexpected token :

Missing Mark Bug

Explain

- **Automatic Semicolon Insertion (ASI)**
  - JavaScript needs semicolons in order to parse the language, however, there is a mechanism called automatic semicolon insertion to assist with parsing http://es5.github.io/#x7.9
- **ASI Rules**
  - Applied when new line or curly brace is followed by invalid token
  - Applied when new line comes before -- or ++ token
  - Applied when new line follows a continue, break, return or throw statement
  - Applied at end of a file if needed to parse
- **ASI Exceptions**
  - Not applied if would result in an empty statement
  - Not applied inside head of a for statement

Missing Mark Bug

Explain

```
function getNames() {
  var length = 0, names = ""

  ['John', 'Susan', 'Joe'].forEach(function (name, i) {
    length = i + 1;
    names += name + ' ';
  });

  return;
  {
    length: length,
    names: names
  };
}
```

Automatic Semicolon Insertion (ASI)

Missing Mark Bug

Explain

Returns `undefined`

```
func
  var length = 0; names = ""['John', 'Susan',
'Joe'].forEach(function (name, i) {
    length = i + 1;
    names += name + ' ';
  });

  return;
  {
    length: length,
    names: names
  };
}
```

Returns `undefined`

Missing Mark Bug

Exterminate

Semicolon-less JavaScript Rules

1. Don't end your statements with a semicolon
2. If statement starts with `[`, `(`, or a binary operator (+*/-,.) then insert a semicolon before it

Example:

```
function bootstrap(home) {
  var selector = typeof home === "string" ?
    "#home" + home : null
  if (selector) home = null
  ;(home || new HomeView(selector)).render()
}
```

Missing Mark Bug


```
function getNames() {
  var length = 0, names = ""

  ;['John', 'Susan', 'Joe'].forEach(function (name, i) {
    length = i + 1
    names += name + ' '
  })

  return {
    length: length,
    names: names
  }
}
```

Missing Mark Bug

Using Semicolons As Expected

1. Follow the official specification when semicolons are required
2. Use tools like JSLint or JSHint to give you feedback and integrate into your code editor

Example:

```
function bootstrap(home) {
  var selector = typeof home === "string" ?
    "#home" + home : null;
  if (selector) home = null;
  (home || new HomeView(selector)).render();
}
```

Missing Mark Bug


```
function getNames() {
  var length = 0, names = "";

  ['John', 'Susan', 'Joe'].forEach(function (name, i) {
    length = index + 1;
    names += name + ' ';
  });

  return {
    length: length,
    names: names
  };
}
```

Fresh Function Bug

Examine

```
var people = [
  { fname: "John", lname: "Smith", bday: "2/2/1979" },
  { fname: "Jane", lname: "Smith", bday: "3/3/1981" },
  { fname: "Jack", lname: "Smith", bday: "4/4/1982" }
];

people.filter(function (person) {
  return new Date(person.bday).getFullYear() < 1980;
}).map(function (person) {
  return {
    name: person.fname + " " + person.lname,
    age: moment().diff(moment(person.bday), "years")
  };
});
```

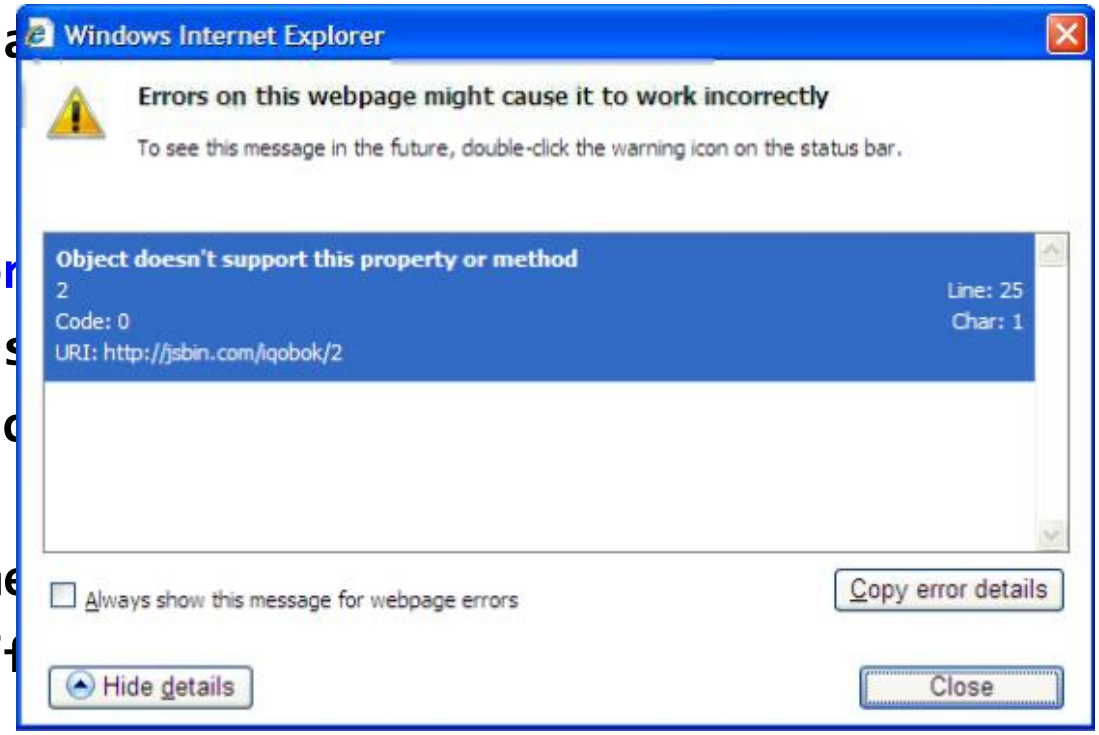
Fresh Function Bug

Expose

```
var people = [
  { fname: "John", lname: "Smith", bday: "2/2/1979" },
  { fname: "Jane", lname: "Smith", bday: "3/3/1981" },
  { fname: "Jack", lna
```

IE8 Throws an Error

```
people.filter(functio
  return new Date(pers
}).map(function (pers
  return {
    name: person.fname
    age: moment().diff
  };
});
```



Fresh Function Bug

Explain

- **ECMAScript 5 array methods in Chrome, Firefox, Safari, Opera, IE9+**
  - map, reduce, reduceRight, filter, forEach, every, some, indexOf, lastIndexOf
- **Polyfill**
  - es5-shim - https://github.com/kriskowal/es5-shim/
- **Shim**
  - Underscore.js - http://underscorejs.org/
  - Lo-Dash - http://lodash.com/

Fresh Function Bug

Exterminate

```
<!DOCTYPE html>
<html xmlns="http://
<head><title>Extermi
<body>
  <script src="es5-shim.min.js"></script>
  <script>
    var people = [ /* ... */ ];

    people
      .filter(function (person) { return /* ... */; })
      .map(function (person) { return /* ... */; });
  </script>
</body>
</html>
```

Polyfilling ECMAScript 5 array methods

Fresh Function Bug

Extra

```
var people = [
  { fname: "John", lname: "Smith", bday: "2/2/1979" },
  { fna                                          "3/3/1981" },
  { fna                                          "4/4/1982" }
];
people.reduce(function (memo, person) {
  if (new Date(person.bday).getFullYear() < 1980) {
    memo.push({
      name: person.fname + " " + person.lname,
      age: moment().diff(moment(person.bday), "years")
    });
  }
  return memo;
}, []);
```

.reduce() can combine .filter() and .map()

Tumble Through Bug

Examine

```
function getPrice(item) {
    var price = 0;
    switch (item) {
        case "apple": price = 1.25; break;
        case "banana": price = 0.75; break;
        case "orange": price = 1; break;
        case "passionfruit": price = 1.5;
        case "pear": price = 0.5; break;
        default: price = 0;
    }
    return price;
}

console.log(getPrice("passionfruit"));
```

Tumble Through Bug

Expose

```
function getPrice(item) {
    var price = 0;
    switch (item) {
        case "apple": price = 1.25; b
        case "banana": price = 0.75;
        case "orange": price = 1; br
        case "passionfruit": price = 1.5;
        case "pear": price = 0.5; break;
        default: price = 0;
    }
    return price;
}

console.log(getPrice("passionfruit")); // 0.5
```

No `break` so falls through to "pear" price

JSHint: Line 7: case "passionfruit": price = 1.50;
--- Expected a 'break' statement before 'case'.

Tumble Through Bug

Explain

```
switch (item) {
  case "value1": /* code */ break;
  case "value2": /* code */ break;
  case "value3": /* code */ break;
  default:       /* code */
}
```

```
switch (item) {
  case "value1": /* co
  case "value2": /* co
  /* falls through */
  case "value3": /* code */ break;
  default:       /* code */
}
```

JSHint won't complain if you add
/* falls through */

Tumble Through Bug

Exterminate

```
function getPrice(item) {
    var price = 0;
    switch (item) {
        case "apple": price = 1.25; break;
        case "banana": price =
        case "orange": price =
        case "passionfruit": price = 1.5; break;
        case "pear": price = 0.5; break;
        default: price = 0;
    }
    return price;
}

console.log(getPrice("passionfruit")); // 1.5
```

Add `break` for "passionfruit"

Tumble Through Bug

Extra

```
var store = (function () {
  var prices = {
    apple: 1.25,
    banana: 0.75,
    orange: 1.0,
    passionfruit: 1.5,
    pear: 0.5
  }, getPrice = function (item, quantity) {
    return prices[item] * quantity;
  };
  return { getPrice: getPrice };
}());

console.log(store.getPrice("passionfruit", 2)); // 3
```

Keep price in an object and encapsulate in a module

Tumble Through Bug

Extra

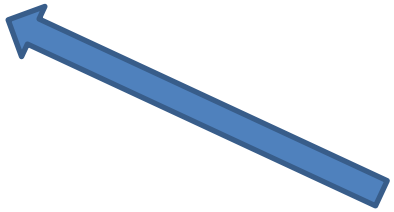
Each fruit has their own function

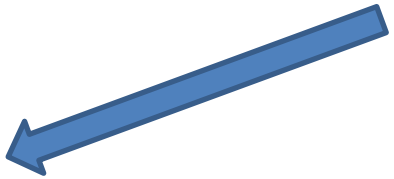
```
var prices = {};

prices.apple  = function (num) { return 1.25 * num; };
prices.banana = function (num) { return 0.75 * num; };
prices.orange = function (num) { return 1.00 * num; };
prices.passionfruit = function (num) {
    var month = new Date().getMonth(),
        price = month < 4 && month > 10 ? 2.5 : 1.5;
    return price * num;
};
prices.pear   = function (num) { return 0.50 * num; };

console.log(prices["passionfruit"](2)); // 3 or 5
```

Strictly Stray Bug

Examine

```
// script1.js
"use strict";
var person = {
  fname: "John",
  lname: "Smith"
};

// script2.js
(function () {
  oops = "uhh-ohh";
  console.log(oops);
}());
```





Two Script Files

Strictly Stray Bug

Expose

```
"use strict";var
person={fname:"John",lname:"Smith"};(function(){oops="uhh
-ohh",console.log(oops)})();
//@ sourceMappingURL=bundle1.min.js.map
```

Error only when combined & minified

Uncaught ReferenceError: oops is not defined

Strictly Stray Bug

Explain

What is Strict Mode?

- Prevent accidental global variables
- Assignment to non-writable property will throw exception
- Deleting undeletable property will throw exception
- Requires properties to be unique in object literal
- Requires parameter names in function to be unique
- Prevent octal number literals
- Makes use of `with` a syntax error
- New variables aren't introduced in surrounding scope with `eval`
- Etc…

Strictly Stray Bug

Explain

Apply Strict Mode

JSHint: "use strict"; --- Use the function form of "use strict".

1. For Scripts

```
"use strict";
var person = { fname: "John" };
console.log("I'm Strict");
```

2. For Functions

```
function test1() {
  "use strict";
  console.log("I'm Strict");
}
function test2() { console.log("I'm Not Strict"); }
```

Strictly Stray Bug

Exterminate

```
// script1.js
(function () {
  "use strict";
  var person = {
    fname: "John",
    lname: "Smith"
  };
}());

// script2.js
(function () {
  oops = "uhh-ohh";
  console.log(oops);
}());
```

Wrap code in IFFE so "use strict"; contained to function

Fickle Figure Bug

Examine

```
$(document).ready(function() {
  $("input").datepicker({
    minDate: -20,
    defaultDate: "+1w",
    maxDate: "+1M +5D",
    showWeek: true,
    numberOfMonths: 3,
  });
});
```

Fickle Figure Bug

Expose

```
$(document).ready(function() {
  $("input").datepicker({
    minDate: -20,
    defaultDate: "+1w",
    maxDate: "+1M +5D",
    showWeek: true,
    // numberOfMonths: 3,
  });
```

IE7 - Error: Expected identifier, string or number

```
$(document).ready(...
  $("input").datepicker({
    minDate: -20
    , defaultDate: "+1w"
    , maxDate: "+1M +5D"
    , showWeek: true
    , numberOfMonths: 3
  });
});
```

Fickle Figure Bug

Explain

Internet Explorer 7

```
var opts = { minDate: -20, showWeek: true, }; // Error
var numbers = [ 1, 2, 3, ]; // Error
```

Internet Explorer 8

```
var opts = { minDate: -20, showWeek: true, }; // Works
var numbers = [ 1, 2, 3, ]; // length 4
```

Internet Explorer 9

```
var opts = { minDate: -20, showWeek: true, }; // Works
var numbers = [ 1, 2, 3, ]; // length 3
```

Fickle Figure Bug

Exterminate

```
$(document).ready(function () {
  $("input
    minDat
    defaul
    maxDat
    showWe
    numberOfMonths: 3
  });
});
```

Errors:

- Line 7: `numberOfMonths: 3,`

  Extra comma. (it breaks older versions of IE)

Remove trailing comma or don't support IE7 or less

Fickle Figure Bug

Extra

If you do decide to drop IE7 support and use trailing commas keep in mind that…

- JSON.parse() has not changed and does not support trailing commas


```
JSON.parse('{"answer":42,}')
// SyntaxError: Unexpected token }

JSON.parse('[42,]')
// SyntaxError: Unexpected token ]
```

Fixing Common JavaScript Bugs

Expressions & Operators

pluralsight
hardcore developer training

Crude Computation Bug

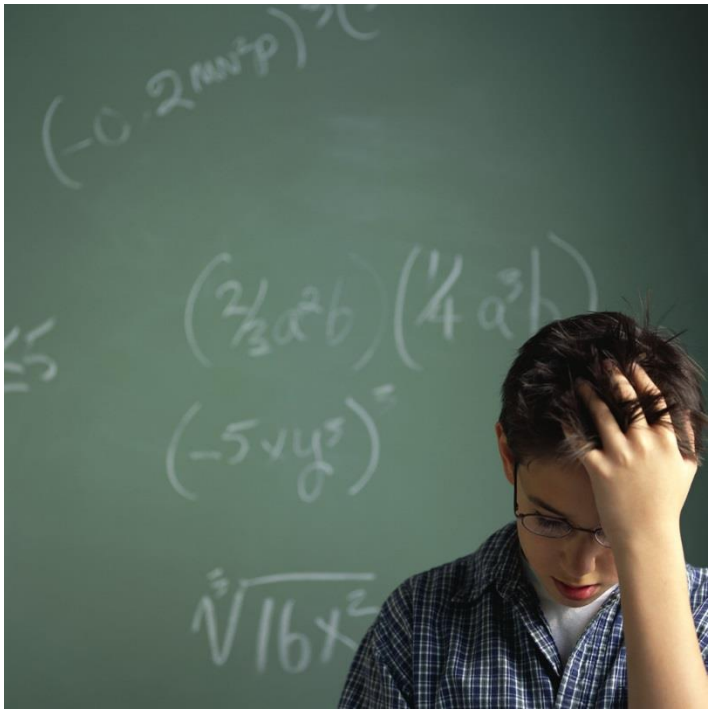
Examine

```
var i;
for (i = 0; i !== 1; i += 0.1) {
  console.log("Hello: " + i);
}
```

Crude Computation Bug

Expose

```
var i;
for (i = 0; i !== 1; i += 0.
  console.log("Hello: " + i)
}
```

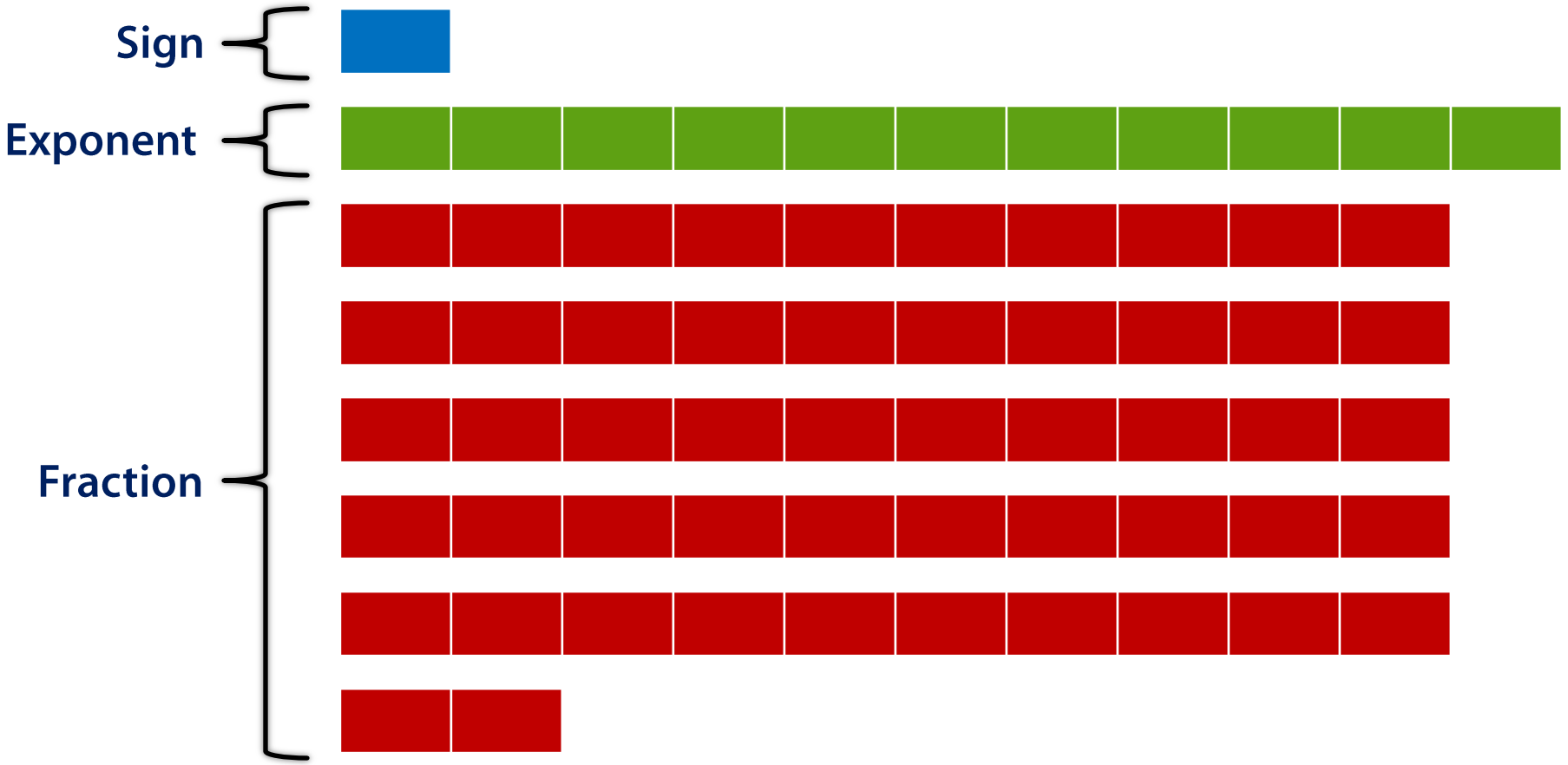


```
Hello: 0
Hello: 0.1
Hello: 0.2
Hello: 0.30000000000000004
Hello: 0.4
Hello: 0.5
Hello: 0.6
Hello: 0.7
Hello: 0.7999999999999999
Hello: 0.8999999999999999
Hello: 0.9999999999999999
Hello: 1.0999999999999999
Hello: 1.2
Hello: 1.3
Hello: 1.4000000000000001
... infinite loop ...
```

Crude Computation Bug

Explain

Number type is IEEE 754 Double Precision floating point



Crude Computation Bug

Explain

```
console.log(0.1 + 0.2);         // 0.30000000000000004
console.log(999999999999999); // 1000000000000000

var money = [1.01, 2.52, 0.77, 1.50, 4.28], sum = 0;
money.forEach(function(i) { sum += i; });
console.log(sum);                   // 10.080000000000002

(function sumMoney(money) {
  var result = 0;
  money = money.map(function(i) { return i * 100 });
  money.forEach(function(i) { result += i; });
  return result / 100;
}(money));                          // 10.08
```

Crude Computation Bug

```
var i;
for (i = 0; i < 1; i += 0.1) {
  console.log("Hello: " + i);
}
```

```
Hello: 0
Hello: 0.1
Hello: 0.2
Hello: 0.30000000000000004
Hello: 0.4
Hello: 0.5
Hello: 0.6
Hello: 0.7
Hello: 0.7999999999999999
Hello: 0.8999999999999999
Hello: 0.9999999999999999
```

Crude Computation Bug


```
var i;
for (i = 0; i < 10; i++) {
  console.log("Hello: " + i / 10);
}
```

```
Hello: 0
Hello: 0.1
Hello: 0.2
Hello: 0.3
Hello: 0.4
Hello: 0.5
Hello: 0.6
Hello: 0.7
Hello: 0.8
Hello: 0.9
```

Crude Computation Bug

Extra

Math Libraries

- Big - https://github.com/MikeMcl/big.js
- BigNumber - https://github.com/MikeMcl/bignumber.js/

```
// Native floating point type
console.log(0.3 - 0.1); // 0.19999999999999998

// BigNumber type
console.log(Big(0.3).minus(0.1).toString()); // "0.2"
```

Behaves more like you might expect

Mistaken Mold Bug

Examine

```
function getResource(url, callbacks) {
  reqwest(url, function (response) {
    if (typeof callbacks === "array") {
      callbacks.forEach(function (cb) { cb(response) });
    } else {
      callbacks(response);
    }
  });
}

function cb1(data) { console.log("callback1", data) }
function cb2(data) { console.log("callback2", data) }
getResource("data.json", cb1);
getResource("data.json", [cb1, cb2]);
```

Mistaken Mold Bug

Expose

```
function getResource(url, callbacks) {
  reqwest(url, function (response) {
    if (typeof callbacks === "array") {
      callbacks.forEach(function (cb) { cb(response) });
    } else {
      callbacks(response);
    }
  });
}
```

```
callback 1 Object { n: "1" }
Uncaught TypeError: object is not a function
```

```
function cb1(data) { console.log("callback1", data) }
function cb2(data) { console.log("callback2", data) }
getResource("data.json", cb1);
getResource("data.json", [cb1, cb2]);
```

Mistaken Mold Bug

Explain

Typeof	Value
`true`	`"boolean"`
`10`	`"number"`
`"Elijah"`	`"string"`
`function () {}`	`"function"`
`undefined`	`"undefined"`
`{}`	`"object"`
`{ name: "John" }`	`"object"`

Typeof	Value
`null`	`"object"`
`new Error()`	`"object"`
`[]`	`"object"`
`[{ name: "John" }]`	`"object"`
`new Date()`	`"object"`
`/^\w$/`	`"object"`

Mistaken Mold Bug

Explain

jQuery	Value
`$.type(null)`	`"null"`
`$.type(new Error())`	`"error"`
`$.type([])`	`"array"`
`$.type([{x:"y"}])`	`"array"`
`$.type(new Date())`	`"date"`
`$.type(/^\w$/)`	`"regexp"`

ECMAScript 5	Value
`Array.isArray([])`	`true`

Underscore Lo-Dash	Value
`_.isNull(null)`	`true`
~~`new Error()`~~	
`_.isArray([])`	`true`
`_.isArray([{x:"y"}])`	`true`
`_.isDate(new Date())`	`true`
`_.isRegExp(/^\w$/)`	`true`
`_.isEmpty({})`	`true`
`_.isArguments(arguments)`	`true`
`_.isFinite(5)`	`true`
`_.isNaN(NaN)`	`true`

Mistaken Mold Bug

Exterminate

```
function getResource(url, callbacks) {
  reqwest(url, function (response) {
    if (_.isArray(callbacks)) {
      callbacks.forEach(function (cb) { cb(response) });
    } else {
      callbacks(response);
    }
  });
}
```

```
callback 1 Object { n: "1" }
callback 1 Object { n: "1" }
callback 2 Object { n: "2" }
```

```
function cb1(data) { console.log("callback1", data) }
function cb2(data) { console.log("callback2", data) }
getResource("data.json", cb1);
getResource("data.json", [cb1, cb2]);
```

Twisted Truth Bug

Examine

```
function sell(item, price) {
  if (price) {
    price = price === 0 ?
      "Free" : "$" + price.toFixed(2);
    console.log("Selling " + item + " for " + price);
  } else {
    console.log("Please provide a price");
  }
}

sell("New Things", 0.50);
sell("Old Things", 0);
sell("Whatchamacallit");
```

Twisted Truth Bug

Expose

```
function sell(item, price) {
  if (price) {
    price = price === 0 ?
      "Free" : "$" + price.toFixed(2);
    console.log("Selling " + item + " for " + price);
  } else {
    console.log("Please provide a price");
  }
}

sell("New Things", 0.50);
sell("Old Things", 0);
sell("Whatchamacallit");
```

```
Selling New Things for $0.50
Please provide a price
Please provide a price
```

Twisted Truth Bug

Explain

The ToBoolean Method (Truthy/Falsey Rules)

Type	Values	Equality
Undefined	undefined	False
Null	null	False
Boolean	false	False
Number	+0, -0, NaN	False
String	"" (Empty)	False
Otherwise		True

FALSEY: `false, 0, -0, null, undefined, NaN, ""`

TRUTHY: `true, 5, "John", {}, [], /^\w+$/, etc...`

Twisted Truth Bug

Exterminate

```
function sell(item, price) {
  if (price !== undefined) {
    price = price === 0 ?
      "Free" : "$" + price.toFixed(2);
    console.log("Selling " + item + " for " + price);
  } else {
    console.log("Please provide a price");
  }
}
sell("New Things", 0.50);
sell("Old Things", 0);
sell("Whatchamacallit");
```

```
Selling New Things for $0.50
Selling Old Things for Free
Please provide a price
```

Crafty Convert Bug

Examine

```
var bacon = {
  slices: 0,
  buy: function (quantity, chocolate) {
    if (quantity == 0) { console.log("WAT?"); }
    if (chocolate == true) { console.log("Adding Joy") }
    this.slices += quantity;
    console.log(this.slices + " total slices of bacon!");
  }
};
bacon.buy(0);
bacon.buy(5);
bacon.buy(10, true);
bacon.buy("", "1");
bacon.buy("!", { toString: function() { return "1" } });
```

Crafty Convert Bug

Expose

```
var bacon = {
  slices: 0,
  buy: function (quantity, chocolate) {
    if (quantity == 0) { console.log("WAT?"); }
    if (chocolate == true) { console.log("Adding Joy") }
    this.slices += quantity;
    console.log(this.slices +
  }
};
bacon.buy(0);
bacon.buy(5);
bacon.buy(10, true);
bacon.buy("", "1");
bacon.buy("!", { toString: fu
```

```
WAT?
0 total slices of bacon!
5 total slices of bacon!
Adding Joy
15 total slices of bacon!
WAT?
Adding Joy
15 total slices of bacon!
Adding Joy
15! Total slices of bacon!
```

Explain

Crafty Convert Bug

The Strict Equality Comparison Algorithm (===)

Type	Values	Equality
Different Types		False
Undefined	Undefined	True
Null	Null	True
Number	Same values (except NaN)	True
String	Same characters	True
Boolean	Both true or both false	True
Object	Both refer to same object	True
Otherwise		False

Note: +0 and -0 are technically different values, but are equal to each other

Explain

Crafty Convert Bug

The Abstract Equality Comparison Algorithm (==)

Type X	Type Y	Equality
Same Types		Strict Equality Comparison Algorithm
Null or Undefined	Null or Undefined	True
Number	String	X == ToNumber(Y)
String	Number	ToNumber(X) == Y
Boolean		ToNumber(X) == Y
	Boolean	X == ToNumber(Y)
String or Number	Object	X == ToPrimitive(Y)
Object	String or Number	ToPrimitive(X) == Y
Otherwise		False

Crafty Convert Bug

Explain

ToNumber Method

Type	Value	Result
Undefined		NaN
Null		+0
Boolean	True	1
Boolean	False	+0
Number		No Conversion
String	""	0
String	"3.2"	3.2
String	"a3.2" or "3.2a"	NaN
Object		ToPrimitive(input); ToNumber(primValue);

Note: ToPrimitive(input) – return valueOf if returns primitive, or toString if returns primitive, otherwise throw and error

Crafty Convert Bug

Explain

The Addition Operator (+)

- Both sides will be toPrimitive(). If either is a String then both sides will be toString'ed and concatenated, otherwise both will be toNumber'ed and added together

```
console.log(4 + 2);           // 6
console.log("4" + 2);         // "42"
console.log("WAT" + 42);      // "WAT42"
console.log("WAT" + "42");    // "WAT42"
console.log("" + 42);         // "42"
```

Crafty Convert Bug

Exterminate

Errors:

- Line 9: `if (quantity == 0) { console.log("WAT?"); }`

  Expected '===' and instead saw '=='.

- Line 10: `if (chocolate == true) { console.log("Adding Joy"); }`

  Expected '===' and instead saw '=='.

```
var
  bu
                                                          ) }
    console.log(this.slices +   total slices of bacon!");
  }
};
bacon.buy(0);
bacon.buy(5);
bacon.buy(10, true);
bacon.buy("", "1");
bacon.buy("!", { toString: function () { return "1" } });
```

```
WAT?
0 total slices of bacon!
5 total slices of bacon!
Adding Joy
15 total slices of bacon!
```

Crafty Convert Bug

Extra

```
var bacon = {
  slices: 0,
  buy: function (quantity, chocolate) {
    if (typeof quantity == "number") {
      if (quantity == 0) { console.log("WAT?") }
      if (typeof chocolate == "boolean" && chocolate) {
        console.log("Adding Joy");
      }
      this.slices += quantity;
      console.log(this.slices +
        " total slices of bacon!");
    }
  }
};
```

Ignored Invention Bug

Examine

```
function Cat(name, breed, color) {
  this.name = name || "Unknown";
  this.breed = breed || "Unknown";
  this.color = color || "Unknown";
}

var fluffy = new Cat("Fluffy", "Ragamuffin", "White"),
  midnight = Cat("Midnight", "Bombay", "Black");

console.log(JSON.stringify(fluffy));
console.log(JSON.stringify(midnight));
```

Ignored Invention Bug

Expose

```
function Cat(name, breed, color) {
  this.name = name || "Unknown";
```

```
{"name":"Fluffy","breed":"Ragamuffin","color":"White"}
undefined
Midnight Unknown Black
```

```
var fluffy = new Cat("Fluffy", "Ragamuffin", "White"),
  midnight = Cat("Midnight", "Bombay", "Black");

console.log(JSON.stringify(fluffy));
console.log(JSON.stringify(midnight));
console.log(window.name, window.breed, window.color);
```

Ignored Invention Bug

Explain

The new Operator

- Creates a new object
- Sets the object's prototype to the constructor function's prototype
- Executes the constructor function passing the new object as its this context
- Returns the newly created object if constructor doesn't return an object

```
function Person(name) {
  this.name = name;
}

var person1 = Person();

var person2 = Person();
```

Refers to global object

Creates new object

Object not created

Considered best practice to upper-case your constructor function to signify to developer that it needs to be new'ed up

Ignored Invention Bug

Explain

Make Your Constructor Function Smarter

- If didn't use new, then call it manually

```
function Person(name) {
    if (!(this instanceof Person)) {
        return new Person(name);
    }
    this.name = name;
}

var person1 = new Person();

var person2 = Person();
```

This technique is meant to protect against accidental creation of objects without using new

Ignored Invention Bug

Errors:

- Line 12: `midnight = Cat("Midnight", "Bombay", "Black");`

  Missing 'new' prefix when invoking a constructor.

```
{"name":"Fluffy","breed":"Ragamuffin","color":"White"}
{"name":"Midnight","breed":"Bombay","color":"Black"}
result undefined undefined
```

```
funct
  if
    r
  }
}
var fluffy = new Cat("Fluffy", "Ragamuffin", "White"),
  midnight = new Cat("Midnight", "Bombay", "Black");

console.log(JSON.stringify(fluffy));
console.log(JSON.stringify(midnight));
console.log(window.name, window.breed, window.color);
```

Fixing Common JavaScript Bugs

Functions

Confounding Context Bug

Examine

```
var student = {
  name: "John Smith",
  resume: [],
  study: function (item) {
    console.log(this.name + " is studying " + item);
    function addToResume(item) { this.resume.push(item) }
    addToResume(item);
  }
}, memorize = student.study;

student.study("chemistry");
console.log(student.resume);
memorize("history");
console.log(student.resume);
```

Confounding Context Bug

Expose

```
var student = {
  name: "John",
  resume: [],
  study: function (item) {
    console.log(this.name + " is studying " + item);
    function addToResume(item) { this.resume.push(item) }
    addToResume(item);
  }
}, memorize = student.study;

student.study("chemistry");
console.log(student.resume);
memorize("history");
console.log(student.resume);
```

```
John is studying chemistry
```

Uncaught TypeError: Cannot call method 'push' of undefined

Confounding Context Bug

Explain

```
function hiWindow() { console.log(this); }
hiWindow(); // [object Window] => global

var person = {
    name: "John Smith",
    greet: function () {
        console.log(this);
    }
};
person.greet(); // [object Object] => person

var hello = person.greet;
hello(); // [object Window] => global
```

Confounding Context Bug

Explain

```
function hiWindow() { console.log("Hello " + this.name) }
hiWindow.call({ name: "John" }); // Hello John

var person = {
  name: "Jane",
  greet: function() { console.log("Hello " + this.name) }
};
var hello = person.greet.bind({ name: "Jake" });
hello(); // Hello Jake

var Person = function(name) { this.name = name; };
Person.prototype.greet =
  function() { console.log("Hello " + this.name) }
new Person("Jane").greet(); // Hello Jane
```

```
hiWindow.apply({ name: "John" })
```

Confounding Context Bug

Explain

Code	this
`this`	window
`myFunction()`	window
`myObject.method()`	myObject
`myFunction.call(context, arg1, arg2)`	context
`myFunction.apply(context, [arg1, arg2])`	context
`var f = myFunction.bind(context); f();`	context
`var myObject = new Object();`	myObject

Confounding Context Bug

```
var student = {
  name: "John", resume: [],
  study: function (item) {
    var that = this;
    console.log(this.name + " is studying " + item);
    function addToResume(item) { that.resume.push(item) }
    addToResume(item);
  }
}, memorize = student.study;

student.study("chemistry");
console.log(student.resume);
memorize.call(student, "history");
console.log(student.resume);
```

```
John is studying chemistry
["chemistry"]
John is studying history
["chemistry", "history"]
```

Confounding Context Bug

Extra

(func
  "us
  fun
  hil

Errors:

- Line 9: `function addToResume(item) { this.resume.push(item); }`

  Possible strict violation.

```
  var person = {
    name: "Jane",
    greet: function () {
      console.log(this);
    }
  };
  person.greet.call(null); // null
}());
```

"use strict"; eliminates this coercion to the global object

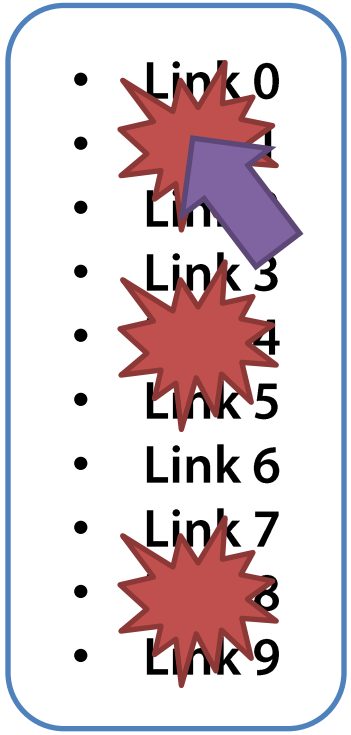
Escaped Environment Bug

Examine

```
var ul = document.querySelector("ul"), i, li;
for (i = 0; i < 10; i++) {
  li = document.createElement("li");
  li.innerHTML = "Link " + i;
  li.addEventListener("click", function () {
    console.log("You've clicked " + i);
  }, false);
  ul.appendChild(li);
}
```

Escaped Environment Bug

```
var ul = document.querySelector("ul"), i, li;
for (i = 0; i < 10; i++) {
  li = document.createElement("li");
  li.innerHTML = "Link " + i;
  li.addEventListener("click", function () {
    console.log("You've clicked " + i);
  }, false);
  ul.appendChild(li);
}
```




```
You've clicked 10
You've clicked 10
You've clicked 10
```

Expose

Escaped Environment Bug

Explain

- We need to introduce the concept of a closure

"A closure is a special kind of object that combines two things: a function, and the environment in which that function was created." --MDN

```
function makeAdder(x) {
    return function (y) { return x + y; };
}

var add5  = makeAdder(5);
var add10 = makeAdder(10);

console.log(add5(2));  // 7
console.log(add10(2)); // 12
```

Escaped Environment Bug

Exterminate

```
var ul = document.querySelector("ul"), i, li;
for (i = 0; i < 10; i++) {
  li = document.createElement("li");
  li.innerHTML = "Link " + i;
  li.addEventListener("click", (function (index) {
    return function () {
      console.log("You've clicked " + index);
    };
  }(i)), false);
  ul.appendChild(li);
}
```

JSHint: Line 9: }(i)), false);
Don't make functions within a loop.

```
You've clicked 1
You've clicked 4
You've clicked 8
```

Escaped Environment Bug

Extra

```
var ul = document.querySelector("ul"), i, li;
for (i = 0; i < 10; i++) {
  li = document.createElement("li");
  li.innerHTML = "Link " + i;
  li.addEventListener("click", clickHandler(i), false);
  ul.appendChild(li);
}

function clickHandler(index) {
  return function() {
    console.log("You've clicked " + index);
  };
}
```

Fixing Common JavaScript Bugs

Values, Variables, and Literals

pluralsight
hardcore developer training

Booked Byword Bug

Examine

```
var collection = (function() {
  var items = [];
  var add = function(item) { items.push(item) };
  var get = function(index) { return items[index] };
  var delete = function(index) { items.splice(index, 1) };

  return {
    add: add,
    get: get,
    delete: delete
  };
}());
```

Booked Byword Bug

Expose

```
var collection = (function() {
  var items = [];
  var add = function(item) { items.push(item) };
  var get = function(index) { return items[index] };
  var remove = function(index) { items.splice(index, 1) };

  return {
    add:
    get:
    delete: remove
  };
}())
```

Uncaught SyntaxError:
Unexpected token delete

Expected identifier, string or number

Booked Byword Bug

Explain

Reserved Keywords

break	case	catch	continue	debugger
default	delete	do	else	finally
for	function	if	in	instanceof
new	return	switch	this	throw
try	typeof	var	void	while
with				

Reserved Keywords for the Future

class	enum	export	extends	implements
import	interface	let	package	private
protected	public	static	super	yield

Booked Byword Bug

Explain

In ECMAScript 3 reserved words can not be used as identifier names or identifiers.

```
// Identifier Names
a.impo
a["imp
a = {
```

Errors:

- Line 8: `var delete = function(index) { items.splice(index, 1); };`

  Expected an identifier and instead saw 'delete' (a reserved word).

```
}
```

```
// Identifier
function import() {}
var import = "test";
```

In ECMAScript 5 reserved words can be used as identifier names, but not identifiers.

```
// Identifier Names
```

```
// Identifier
function import() {}
var import = "test";
```

Booked Byword Bug


```
var collection = (function() {
  var items = [];
  var add = function(item) { items.push(item) };
  var get = function(index) { return items[index] };
  var remove = function(index) { items.splice(index, 1) };

  return {
    add: add,
    get: get,
    "delete": remove
  };
}());
```

```
collection["delete"](1);
```

Relative Realism Bug

Examine

```
var element = document.getElementById("greeting");
function html(value) {
    if (value === undefined) {
        return element.innerHTML;
    } else if (typeof value === "string") {
        element.innerHTML = value;
    } else if (typeof value === "function") {
        element.innerHTML = value(element.innerHTML);
    }
}
html("Hello");
console.log(html());
html(function (text) { return text + " World!"; });
console.log(html());
```

Relative Realism Bug

Expose

```
undefined = true;
var element = document.getElementById("greeting");
function html(value) {
    if (value === undefined) {
        return element.innerHTML;
    } else if (typeof value === "string") {
        element.innerHTML = value;
    } else if (typeof value === "function") {
        element.innerHTML = value(element.innerHTML);
    }
}
html("Hello");
console.log(html());
html(function (text) { return text + " World!"; });
console.log(html());
```

```
undefined
undefined
```

Relative Realism Bug

Explain

Reserved Keywords

break	case	catch	continue	debugger
default	delete			finally
for	function			instanceof
new	return			throw
try	typeof			while
With				

WHERE'S UNDEFINED!?!

Errors:

- Line 4: `undefined = true;`

  Bad assignment.

Reserved Keyw

class				plements
import				ivate
protected				eld

In ECMAScript 3 undefined was not a reserved word & could be reassigned!

Thankfully in ECMAScript 5 undefined, NaN, & Infinity are all read-only

Relative Realism Bug

Explain

Use IIFE to Protect Undefined & Help with Minification

```
function sayHello(name, empty) {
  console.log("Hi" + name, empty);  // Hi John undefined
}
sayHello("John");

(function (name, undefined) {
  // forcing undefined variable to have undefined value
  if (name === undefined) { console.log("undefined") }
}("John"));

!function (n, o) { n === o && console.log("Name is
undefined") }("John")
```

Relative Realism Bug


```
undefined = true;
(function (undefined) {
  var element = document.getElementById("greeting");
  function html(value) {
    if (value === undefined) {
      return element.innerHTML;
    } else if (typeof value === "string") {
      element.innerHTML = value;
    } else if (typeof value === "function") {
      element.innerHTML = value(element.innerHTML);
    }
  }
  html(function (text) { return text + " World!"; });
  alert(html());
}());
```

Tangled Tag Bug

Examine

```
<!DOCTYPE html>
<html>
<head>
  <!- jquery.js & jquery-ui.js, jquery-ui.css -->
</head>
<body>
  <input id="datePicker" class="date" />
  <script>
    datePicker = $(".date");
    datePicker.datepicker();
  </script>
</body>
</html>
```

Tangled Tag Bug

Expose

```
<!DOCTYPE html>
<html>
<head>
  <!- jquery.js & jquery-ui.js, jquery-ui.css -->
</head>
<body>
  <input id="datePicker" class="date" />
  <script>
    datePicker = $(".date");
    datePicker.d[illegible]nicker();
  </script>
</body>
</html>
```

Object doesn't support this property or method

Tangled Tag Bug

Explain

Named acce…amed-access-window

- Names of …, frameset)
- A, applet, …bject that have a `name` at…
- HTML ele…

Errors:

- Line 1: `datePicker = $(".date");`

  'datePicker' is not defined.

- Line 2: `datePicker.datepicker();`

  'datePicker' is not defined.

```
<div id="h…
<script>
  hello.innerHTML = "Howdy!";
</script>
```

Same thing as…
`window.hello.innerHTML = "Howdy!";`

Tangled Tag Bug

Exterminate

```
<!DOCTYPE html>
<html>
<head><!- jquery.js,jquery-ui.js,jquery-ui.css --></head>
<body>
  <input id="datePicker" class="date" />
  <script>
    (function () {
      var datePicker = $(".date");
      datePicker.datepicker();
    }());
  </script>
</body>
</html>
```

Tangled Tag Bug

Exterminate

```
<!DOCTYPE html>
<html>
<head><!- jquery.js,jquery-ui.js,jquery-ui.css --></head>
<body>
  <input id="datePicker" class="date"></input>
  <script>
    $(document).ready(function () {
      var datePicker = $(".date");
      datePicker.datepicker();
    });
  </script>
</body>
</html>
```

Transform Total Bug

Examine

```
function purchase(item, amount) {
    amount = parseInt(amount);
    console.log("Got " + item + ": $" + amount.toFixed(2));
}

purchase("Eggs", "01");
purchase("Bacon", "08");
```

Transform Total Bug

Expose

```
function purchase(item, amount) {
    amount = parseInt(amount);
    console.log("Got " + item + ": $" + amount.toFixed(2));
}

purchase("Eggs", "01");
purchase("Bacon", "08");
```


```
Got Eggs: $1.00
Got Bacon: $0.00
```

Transform Total Bug

Explain

Usage

```
var p                                    dix);
```

Errors:

- Line 4: `amount = parseInt(amount);`

  Missing radix parameter.

If radix is undefined or 0 then…

- If string & starts with "0x" or "0X" then radix is 16
- If string & starts with "0" then radix is 8 *
- If string & starts with something else then radix 10

* Exception: If ECMAScript 5 then radix is 10

```
console.log(parseInt("0xA")); // 10
console.log(parseInt("015")); // 13 (ES3) or 15 (ES5)
```

Transform Total Bug

Exterminate

```
function purchase(item, amount) {
    amount = parseInt(amount, 10);
    console.log("Got " + item + ": $" + amount.toFixed(2));
}

purchase("Eggs", "01");
purchase("Bacon", "08");
```

```
Got Eggs: $1.00
Got Bacon: $8.00
```

Fixing Common JavaScript Bugs

Objects

Pregnable Property Bug

Examine

```
<!DOCTYPE html>
<html>
<head><script src="./mootools.js"></script></head>
<body>
  <script>
    var customers = ["John Smith", "Susan Smith"], key;

    customers[3] = "Jane Smith";
    for (key in customers) {
      console.log(key, customers[key]);
    }
  </script>
</body>
</html>
```

Pregnable Property Bug

Expose

```
<!DOCTYPE html>
<html>
<head><script                                          /head>
<body>
  <script>
    var custom                                         ith"], key;

    customers[
    for (key i
      console.
    }
  </script>
</body>
</html>
```

```
0 John Smith
1 Susan Smith
3 Jane Smith
$family function()
@constructor [ undefined ]
each function()
clone function()
clean function()
invoke function()
associate function()
link function()
contains function()
...
```

Pregnable Property Bug

Explain

- `for-in` Iterates over [[Enumerable]] properties, including those inherited from the object's prototype chain
- MooTools & Prototype.js add custom prototype methods to Array, String, and others

```
var myArray = ["test1"], name;
myArray[3] = "test2";

Array.prototype.wat = "WAT!?!";
for (name in myArray) { console.log(name, myArray[name]) }
```

```
0 test1
3 test2
wat WAT!?!
```

```
for (var i = 0, len = myArray.length; i < len; i++) {
  console.log(i, myArray[i]);
}
```

```
0 test1
1 undefined
2 undefined
3 test2
```

Pregnable Property Bug

Explain

- for-in works with Arrays and other types, however, it is most valuable when iterating over an object, but be mindful of the prototype

```
function Person(name) { this.name = name; };
Person.prototype.married = false;

var john = new Person("John Smith"), k;
john.married = true;
for (k in john) { console.log(k, john[k]); }
```

```
name John Smith
married true
```

```
var susan = new Person("Susan Smith");
for (k in susan) {
  if (susan.hasOwnProperty(k)) {
    console.log(k, susan[k]);
  }
}
```

```
name Susan Smith
```

Pregnable Property Bug

Exterminate

```
var customers = ["John Smith", "Susan Smith"], key;
customers[3] = "Jane Smith";
for (key in customers) {
  if (customers.hasOwnProperty(key)) {
```

```
0 John Smith
1 Susan Smith
```

Errors:

- Line 4: `for (item in customers) {`

  The body of a for in should be wrapped in an if statement to filter unwanted properties from the prototype.

```
var i, len;
for (i = 0,
     len = customers.length; i < len;
     i++) {
  console.log(i, customers[i]);
}
```

```
0 John Smith
1 Susan Smith
2 undefined
3 Jane Smith
```

Eccentric Envelope Bug

Examine

```
var contestants = ["John Smith", "Jane Smith"];
function isWinner(person) {
  var winner = contestants.some(function (contestant) {
    return person.name === contestant && person.winner;
  });
  if (winner) { console.log(person.name + " :)"); }
  else { console.log(person.name + " :("); }
}

isWinner({ name: new String("Elijah Manor"), winner: new
Boolean(false) });

isWinner({ name: new String("John Smith"), winner: new
Boolean(true) });
```

Eccentric Envelope Bug

Expose

```
var contestants = ["John Smith", "Jane Smith"];
function isWinner(person) {
  var winner = contestants.some(function (contestant) {
    return person.name === contestant && person.winner;
  });
  if (winner) { cons                    e + " :)"); }
  else { console.log                    ("); }
}

isWinner({ name: new String("Elijah Manor"), winner: new
Boolean(false) });

isWinner({ name: new String("John Smith"), winner: new
Boolean(true) });
```

```
Elijah Manor :(
John Smith :(
```

Eccentric Envelope Bug

Explain

JavaScript has 5 Primitive Types

- boolean, number, string, `null`, & `undefined`

JavaScript also has 3 Constructor Wrappers

- `Boolean`, `Number`, & `String`

```
typeof true // boolean
typeof new Boolean(true) // object
new Boolean(true) === new Boolean(true) // false
true == new Boolean(true) // true

typeof "Hello" // string
typeof new String("Hello") // object
new String("Hello") === new String("Hello") // false
new String("Hello") == "Hello" // true
```

Eccentric Envelope Bug

Exterminate

```
var contestants = ["John Smith", "Jane Smith"];
function isWinner(person) {
  va                                                    {
                                                        ;
  })
  if
  el
}

isWi

isWinner({ name: "John Smith", winner: true });
```

Errors:

- Line 12: `isWinner({ name: new String("Elijah Manor"), winner: new Boolean(false) });`
  Do not use String as a constructor.
- Line 12: `isWinner({ name: new String("Elijah Manor"), winner: new Boolean(false) });`
  Do not use Boolean as a constructor.
- Line 14: `isWinner({ name: new String("John Smith"), winner: new Boolean(true) });`
  Do not use String as a constructor.
- Line 14: `isWinner({ name: new String("John Smith"), winner: new Boolean(true) });`
  Do not use Boolean as a constructor.

Eccentric Envelope Bug

Extra

To Boolean

```
console.log(!!"false");              // true
console.log(Boolean("false"));       // true
```

Number To String

```
console.log(String(42));             // "42"
console.log(42 + "");                // "42"
console.log(42..toString());         // "42"
```

String To Number

```
console.log(+"42");                  // 42
console.log(Number("42"));           // 42
console.log(parseInt("42", 10));     // 42
```

Strange Set Bug

Examine

```
var easyCombination = new Array(13),
  hardCombination   = new Array(42, 16, 21),
  combined = easyCombination.concat(hardCombination);

console.log(JSON.stringify(combined));
```

Strange Set Bug

Expose

```
var easyCombination = new Array(13),
  hardCombination   = new Array(42, 16, 21),
  combined = easyCombination.concat(hardCombination);

console.log(JSON.stringify(combined));
```

```
[null, null, null, null, null,
null, null, null, null, null,
null, null, null, 42, 16, 21]
```

Strange Set Bug

Explain

Array Constructor is Overloaded

```
new Array() // []
```

```
new Array(1, 2, 3) // [1, 2, 3]
new Array("1", "2", "3") // [ "1", "2", "3" ]
```

```
new Array(3) // [undefined, undefined, undefined]
```

Strange Set Bug

Exterminate

```
    // JSHint: The array literal notation [] is preferrable.
    // JSLint: Use the array literal notation []
var myArray1 = new Array(),

    // No Warnings
    myArray2 = new Array(5),

    // JSLint: Use the array literal notation []
    myArray3 = new Array(1, 2, 3),

    // JSLint: Use the array literal notation []
    myArray4 = new Array("a", "b", "c");
```

Questions?

Elijah Manor
http://elijahmanor.com
@elijahmanor

pluralsight
hardcore developer training